

FIG. 1

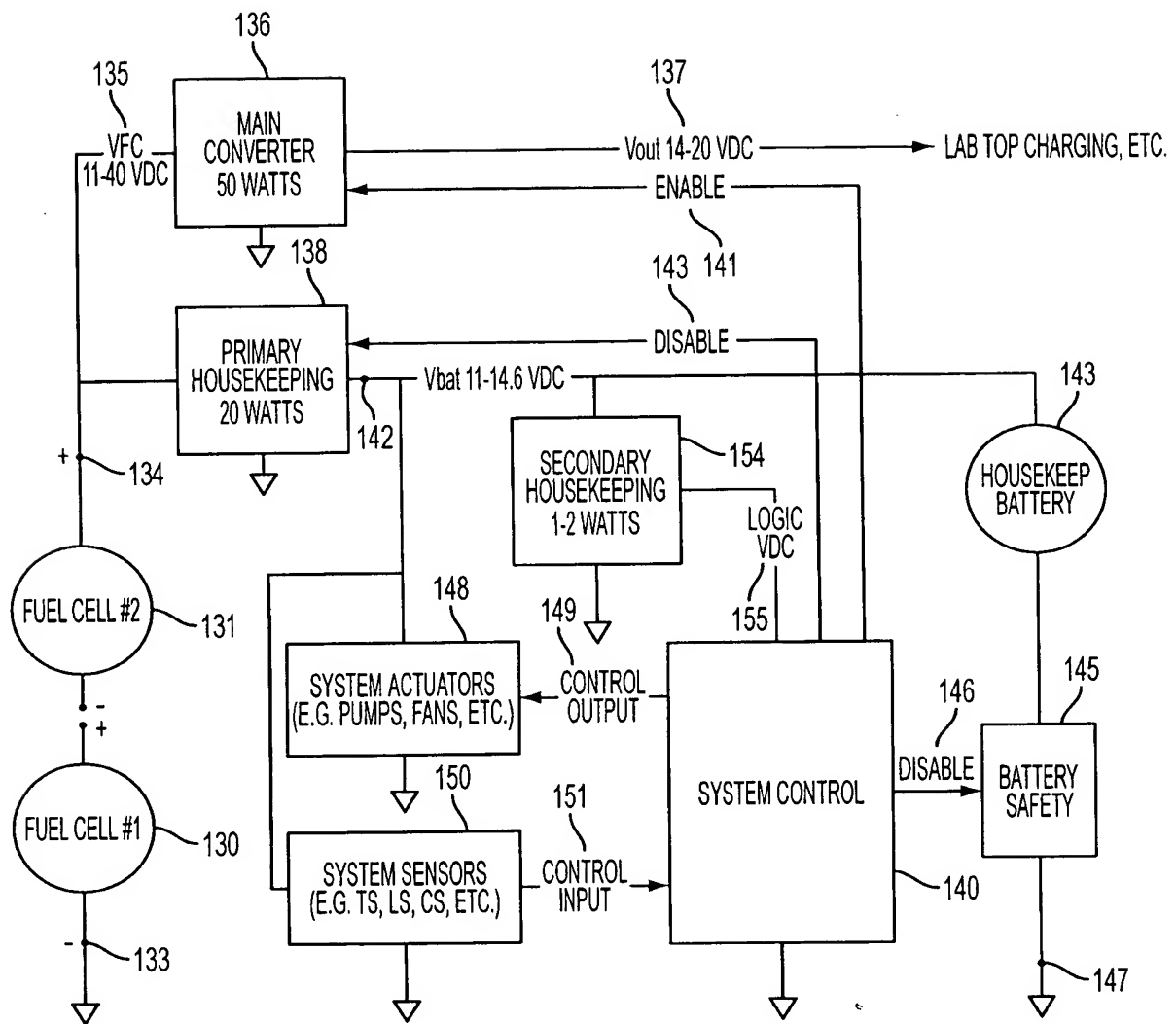


FIG. 2

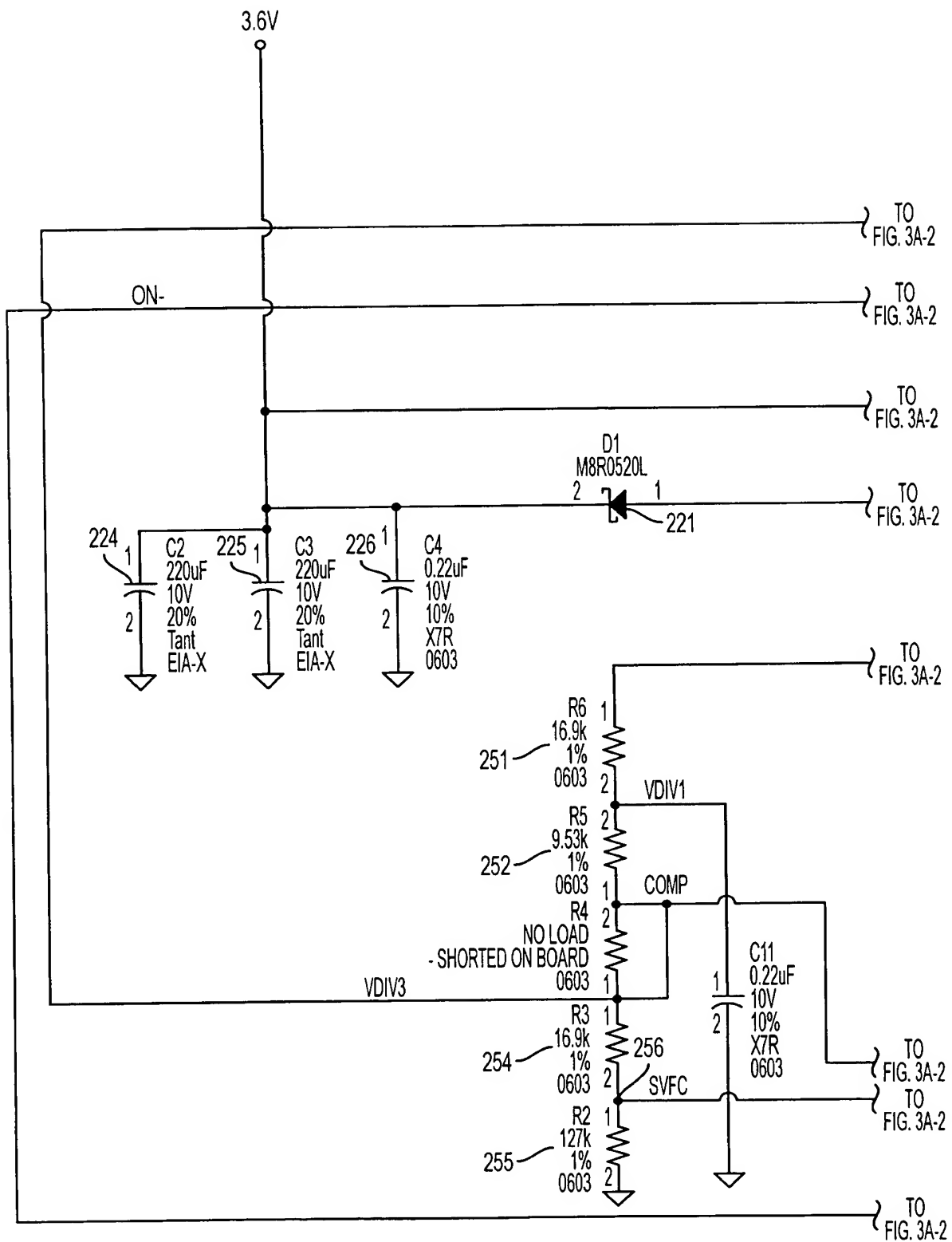


FIG. 3A-1

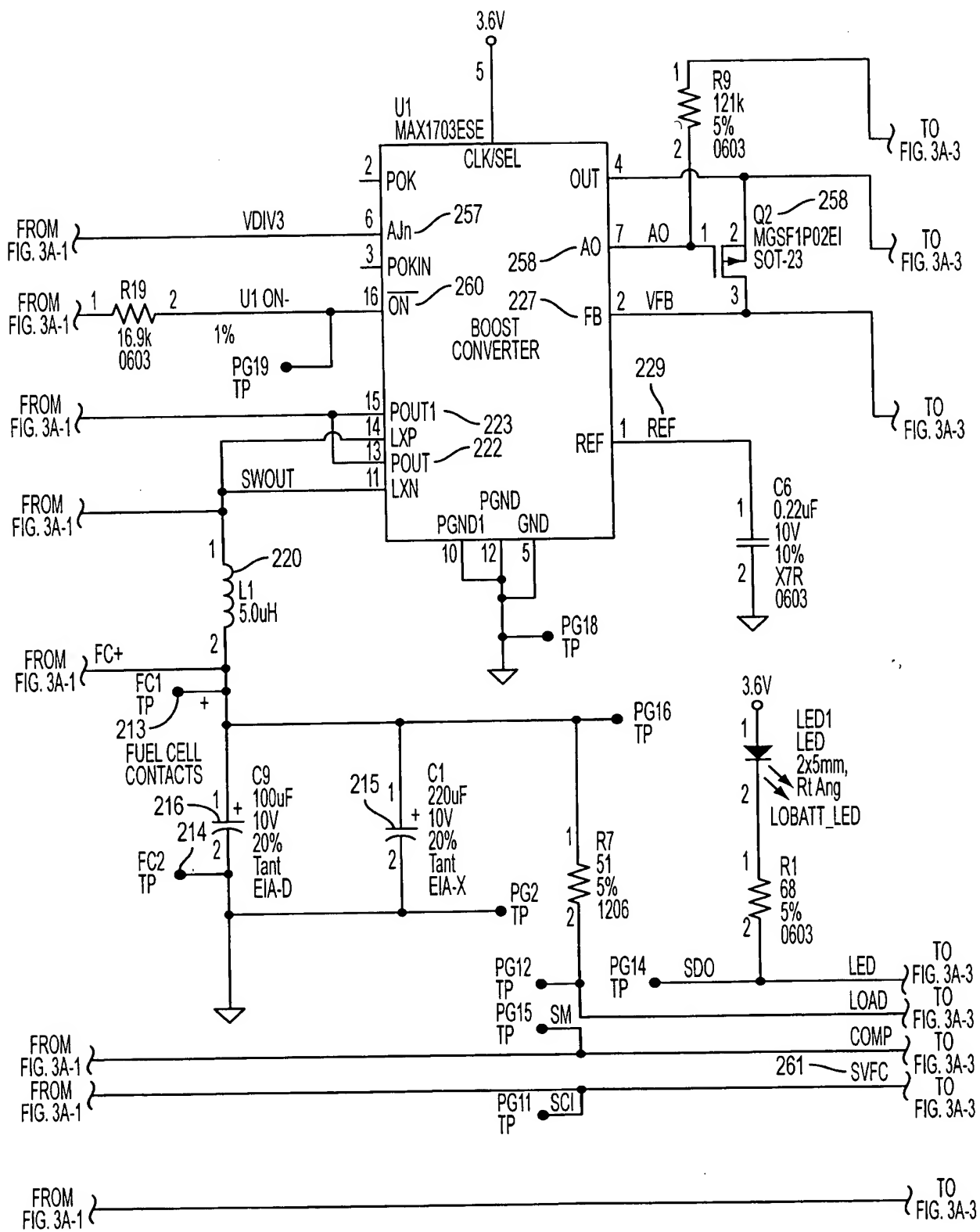
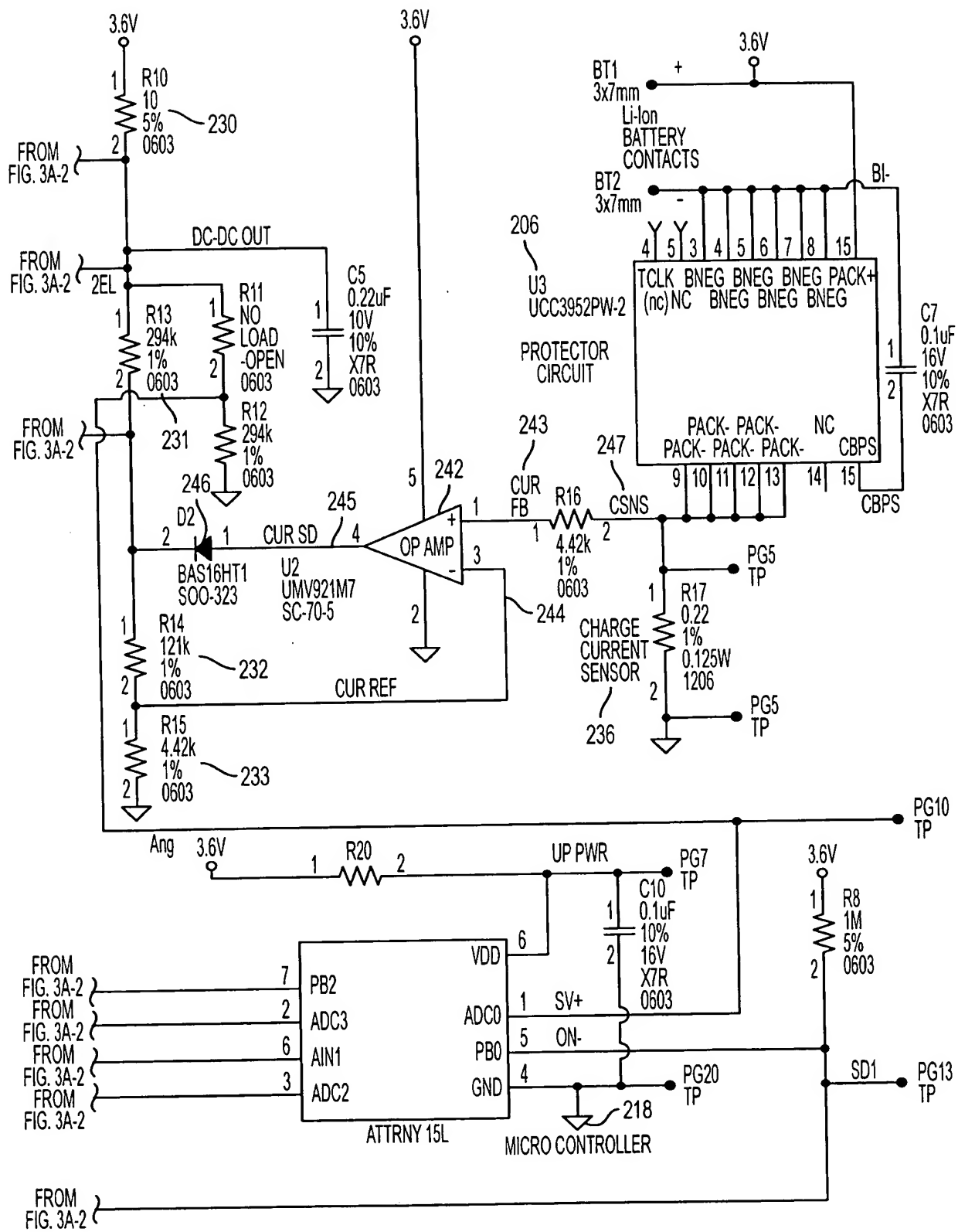


FIG. 3A-2



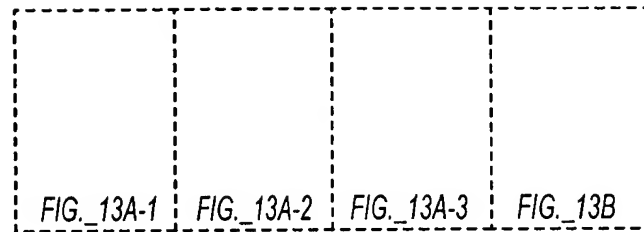


FIG. 3A

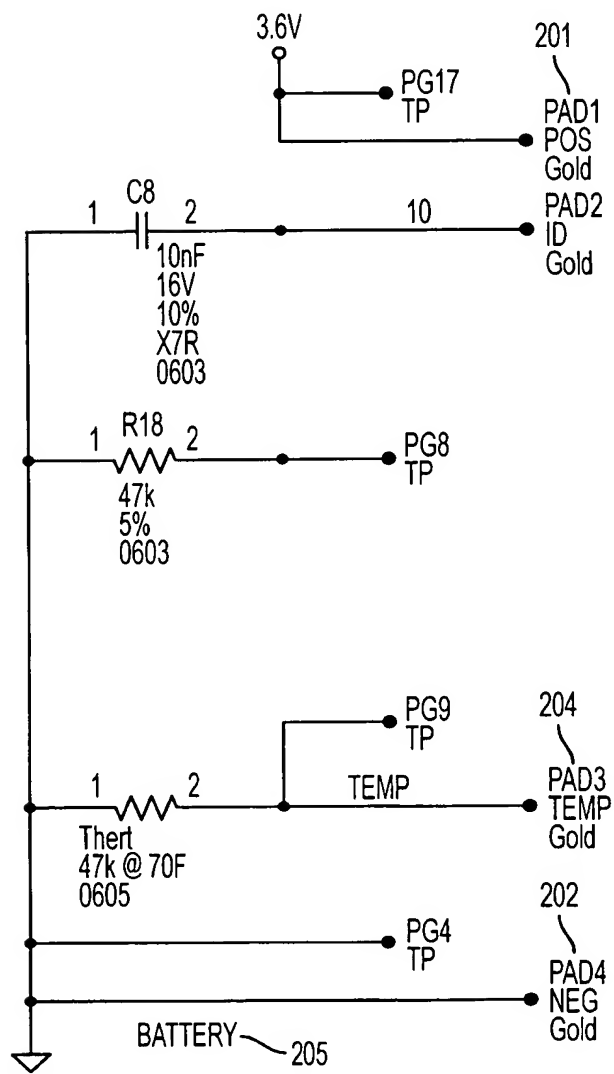


FIG. 3B

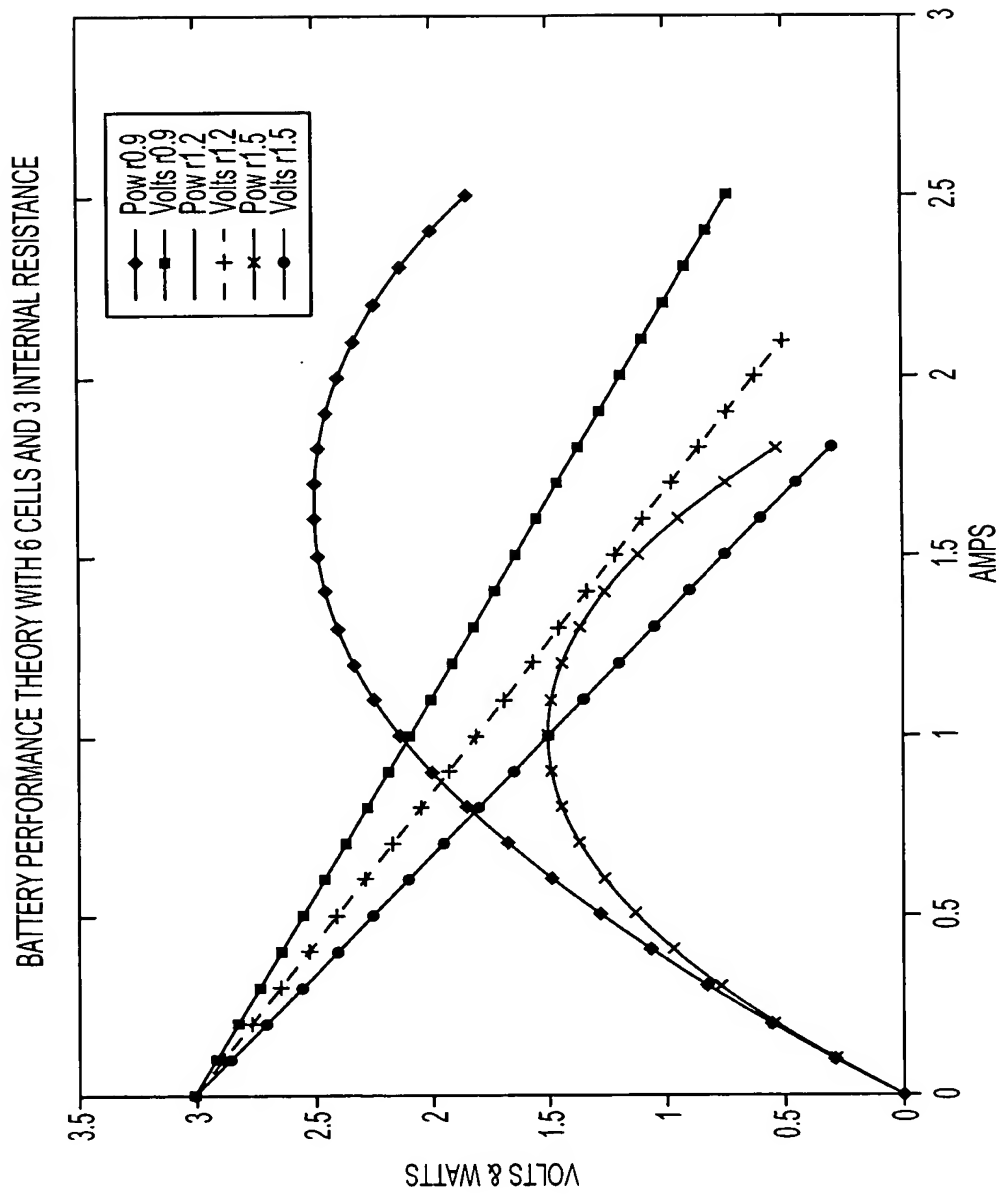


FIG. 4

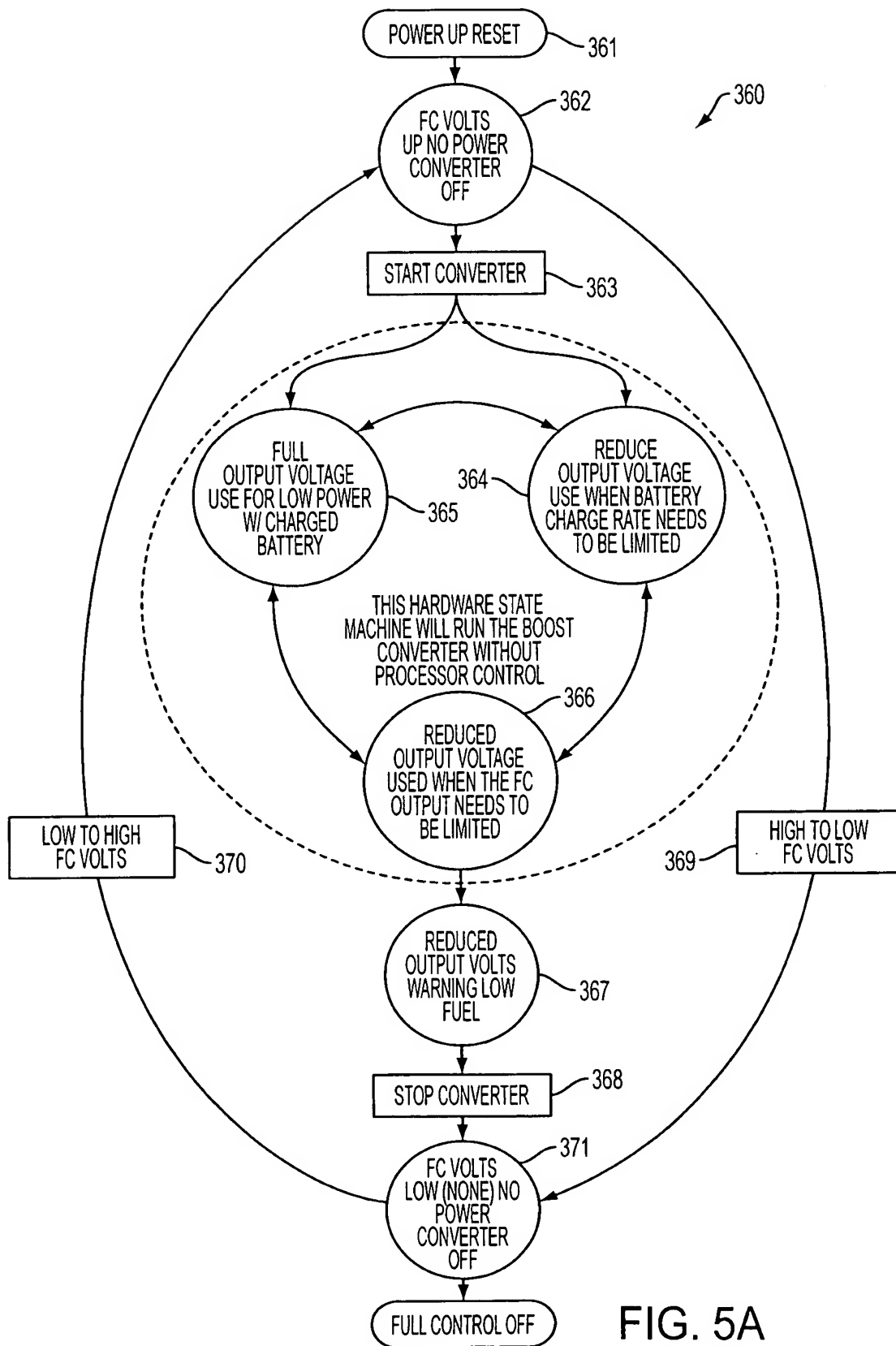


FIG. 5A

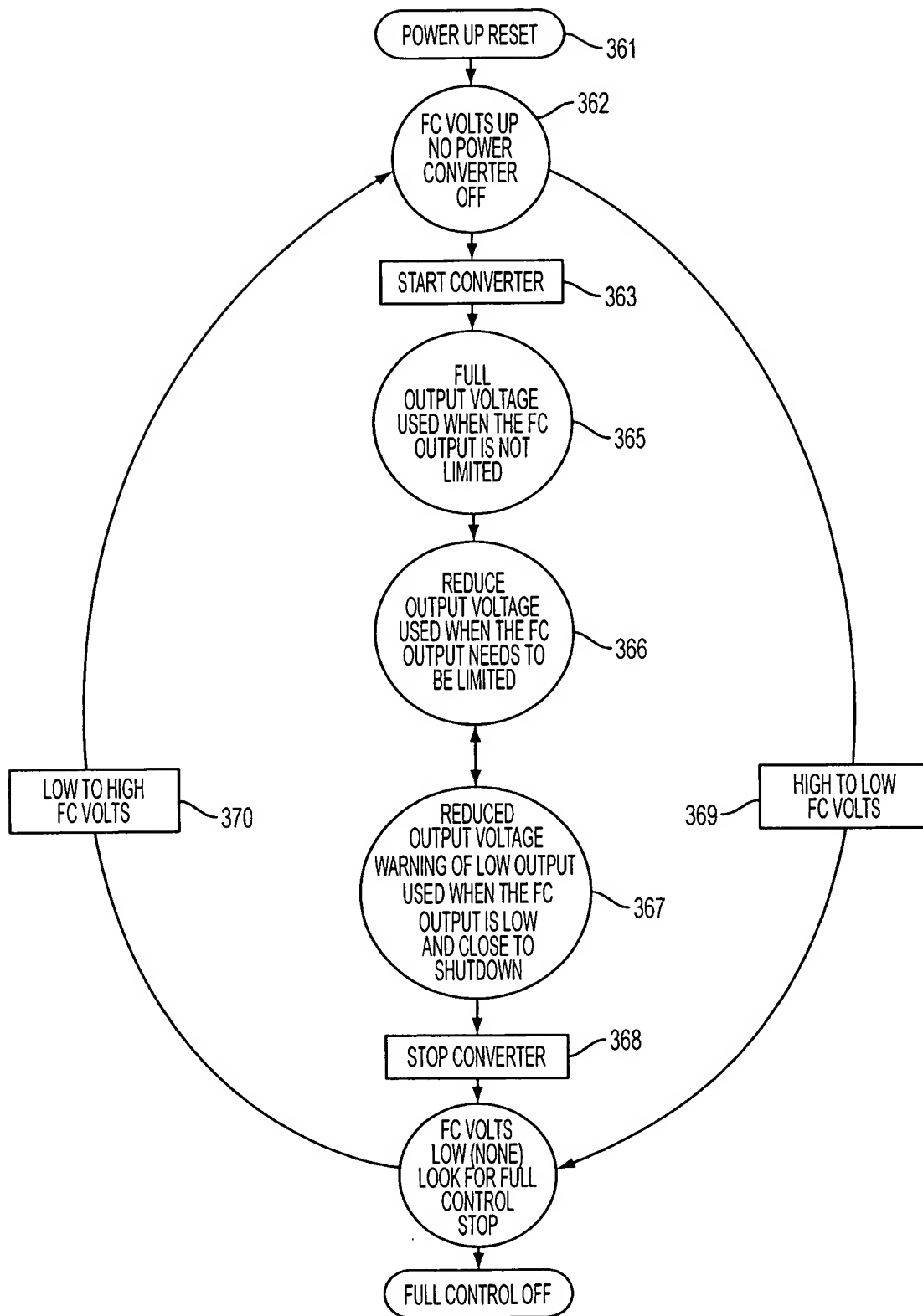


FIG. 5B

SOFTWARE VFC LEVELS

LEVEL 1 - 2.4V IF OFF LOAD TEST ABOVE, WITH WARNING
 LEVEL 2 - 1.5V IF ON SLEEP ABOVE
 LEVEL 3 - 1.2V IF ON WARNING BELOW
 LEVEL 4 - 1.1V IF ON STOP BELOW

HARDWARE TRIP FROM SLEEP IS BETWEEN L2-L3

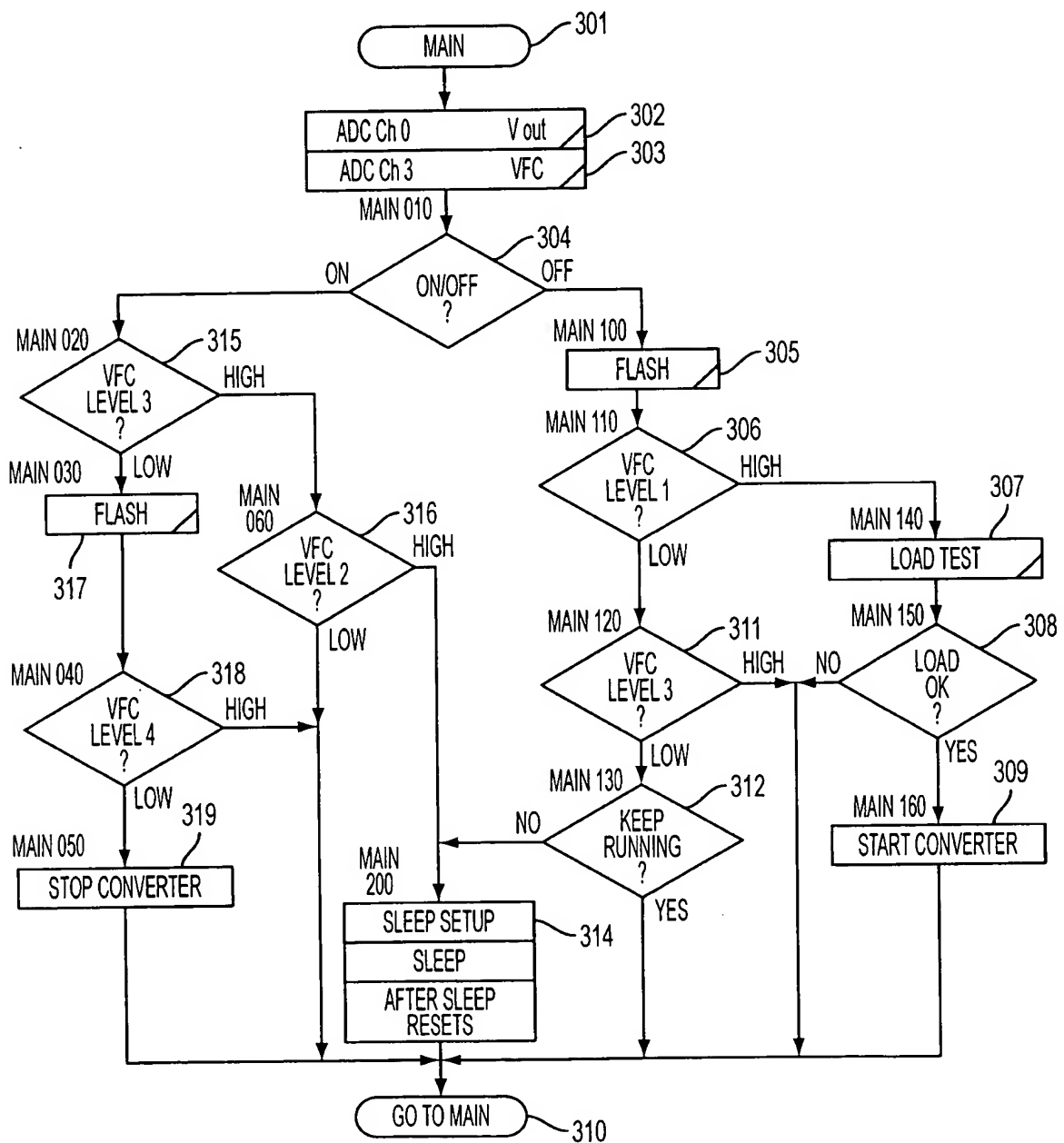


FIG. 6

SOFTWARE VFC LEVELS

LEVEL 1 - 2.4V IF OFF LOAD TEST ABOVE, WITH WARNING
 LEVEL 2 - 1.5V IF ON SLEEP ABOVE
 LEVEL 3 - 1.2V IF ON WARNING BELOW
 LEVEL 4 - 1.1V IF ON STOP BELOW

HARDWARE TRIP FROM SLEEP IS BETWEEN L2-L3

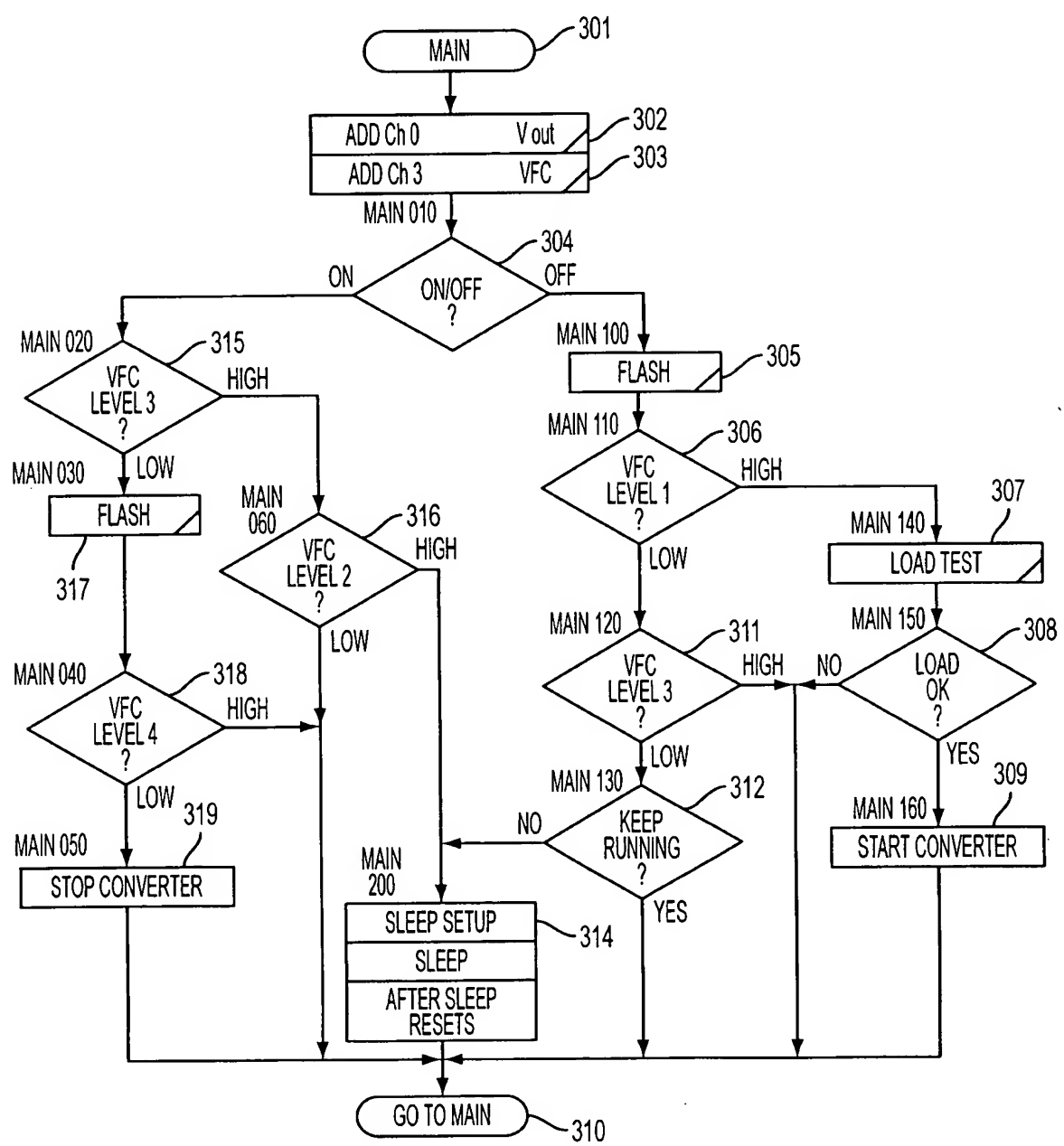


FIG. 7A

MAIN:	<pre> retl ; ADC ; Main Program start ; read adc Ch for SVcc base ldi ZH, SVCC ; set chanel ldi ZL, VCCLOWN ; data store pointer rcall ADCRUN ; call ACD Converter ; read adc ch for SVFC ldi ZH, SVFC ; set chanel ldi ZL, VRCBAN ; data store pointer rcall ADCRUN ; call ACD Converter ldi VFCBAHIGH,\$03 ; ... test for VFC ldi VFCBALOW,\$5e ldi VFCLAHIGH,\$02 ; ... test for VFC ldi VFCLALOW,\$ff cbi ddrB,onnot </pre>
MAIN010:	<pre> ; test OFF/ON sbic PINB,ONNOT ; read ON pin rjmp MAIN100 </pre>
MAIN020:	<pre> ; test vfc for level 3 cpi VFCBALOW,low(LEVEL3) ldi TEMP,high(LEVEL3) cpc VFCBAHIGH,TEMP brsh MAIN060 ; the value is equ or high </pre>
MAIN030:	<pre> rcall FLASH ; do flash </pre>
MAIN040:	<pre> ; test vfc for level 4 cpi VFCBALOW,low(LEVEL4) ldi TEMP,high(LEVEL4) cpc VFCBAHIGH,TEMP brsh MAIN ; the value is equ or high, loop to </pre>
main	<pre> ; rjmp MAIN050 ; the value is low fall or jump </pre>
MAIN050:	<pre> ; Stop the converter sbi PORTB,ONNOT ; Stop Converter and test led rjmp MAIN ; loop to main </pre>
MAIN060:	<pre> ; test vfc for level 2 cpi VFCBALOW,low(LEVEL2) ldi Temp,high(LEVEL2) cpc VFCBAHIGH,TEMP brsh MAIN200 ; the value is equ or high rjmp MAIN ; the value is low fall or jump </pre>
MAIN100:	<pre> ; do flashe rcall FLASH ; do flash </pre>
MAIN110:	<pre> ; test vfc for level 1 cpi VFCBALOW,low(LEVEL1) ldi TEMP,high(LEVEL1) cpc VFCBAHIGH,TEMP brsh MAIN140 ; the value is equ or high ; rjmp MAIN120 ; the value is low fall or jump </pre>
MAIN120:	<pre> ; test vfc for level 3 cpi VFCBALOW, low(LEVEL3) ldi TEMP,high(LEVEL3) cpc VFCBAHIGH,TEMP </pre>

FIG. 7C

```

; the value is equ or high
; the value is low fall or jump
brsh      MAIN
; rjmp    MAIN130

MAIN130:  ; test Flasher for stoped
cpi      FLASHHIGH, STOPED
breq     MAIN200      ; we need to sleep
rjmp     MAIN          ; keep logging

MAIN140:  ; do load test
rcall    LOADTEST     ; test the load

MAIN150:  ; Test for Load OK
lsth     LOADOK
breq     MAIN          ; go to main

MAIN160:  ; start the converter
cbi      PORTB, ONNOT  ; Start Converter
clr      FLASHLOW     ; Stop Flashing
clr      FLASHHIGH
rjmp     MAIN          ; keep looping

MAIN200:  ; enter sleep mode
cbi      ADCSR, ADEN   ; Power down the ADC
clr      TICA
clr      TICB
ldi      TEMP, 0       ; stop timer int
out      TIMSK, TEMP

ldi      TEMP, MCUCRSET ; set for idel
out      MCUCR, TEMP

; may have to stop timers adc interrupts
; sbi     ddrb, led     ; *****
sleep    ; wate COMPARE
; cbi     ddrb, led     ; *****

MAIN210:  ; nop
; nop
; rjmp    MAIN210      ; we will wate to hear for a low level 2

transet

ldi      TEMP, TIMSKSET ; Enable timer int
out      TIMSK, TEMP

sbi      ADSCR, ADEN    ; Power up the ADC
rjmp     MAIN          ; back to looping

; Place init code hear
RESET:

; Clear Requesters
clr      r0             ; Clear a master
ldi      z1, 29         ; Point to req r29
st       z, r0          ; Clear
dec      z1             ; set for next
brne     RESET01        ; loop

; Setup the ADC
ldi      TEMP, ADCSRSET
out      ADCSR, TEMP
sbi      ADCSR, ADEN    ; Power up the ADC

; Setup the comparitor
ldi      TEMP, ACSRSET
out      ACSR, TEMP

; Setup timer 0 for div 64
ldi      TEMP, TCCR0SET

```

FIG. 7D

```

out                TCCR0, TEMP
; Setup Timer 1 for 1.ms int
ldi                TRMP, TCCR1SET
out                TCCR1, TEMP
ldi                TEMP, OCR1ASET
out                OCR1A, TEMP

; Setup Port B
ldi                TEMP, DDRBSET    ; Data direction
out                DDRB, TEMP
ldi                TEMP, PORTBSET
out                PORTB, TEMP

; ldi                ticb, 100

; Enable Interrupts
ldi                TEMP, TIMSKSET    ; Enable timer int
out                TIMSK, TEMP
ldi                TEMP, GIMSKSET    ; Set the mask
out                GIMSK, TEMP
ldi                TEMP, SREGSET     ; Enable
out                SREG, TEMP

; Setup sleep
ldi                TEMP, MCUCRSET
out                MCUCR, TEMP

; setup start delays
ldi                TICA, FLASHDEL    ; flash start delay
ldi                TICB, LOADDEL     ; load start delay

RESETEND:
rjmp               MAIN

; This ISR will dec the Time registers tica and ticb to 0
TIM1_CMP:
in                 TEMPF, SREG        ; save status
tst                tica
breq               tic01
dec                tica
Tic01:
tst                ticb
breq               tic02
dec                ticb
Tic02:
out                SREG, TEMPF        ; Restor status
reti

; This ISR will handal end of time 0 overflows
TIM0_OVF:
; we ret at vector
reti

; This ISR will handal changes in FC Volts it will retem to last place
ANA_COMP:
; we may whant to fix timer for fast service in main
reti

ADC:
reti

EE_RDY
later
reti                ; This ISR may be used

TIM1_OVF:
reti                ; This ISR will be
disabled

```

FIG. 7E

```

; Rutine to manage low fuel flasher
; The two byte flash count also acts as a run flag as follows:
;   Low byte not 0, the counter is active and flashing
;   Low byte equ 0, the high byte has meaning as follows:
;       0 = clear to start flashing
;       1 = flash time complet
;
; any other go to sleep
FLASH:
; Start Flasher
tst     TICA           ; test for time to run
brne    FLASHEND       ; must be zero th run
ldi     TICA, TICFLASH ; reset the timer

tst     FLASHLOW       ; test for need
brne    FLASH10        ; go to flashing
tst     FLASHHIGH      ; test for stoped
brne    FLASHEND       ; the flasher is stoped

; Start the flasher
ldi     FLASHLOW, LOW(FLASHSET)
ldi     FLASHHIGH, HIGH(FLASHSET)

FLASH10:
; flash the LED
cbi     PORTB, LED     ; LED lamp on

; time the flash
ldi     TEMP, TIME40m  ; load time value
rcall   WATE           ; wate for time
;out    TCNT0, TEMP
;ldi    TEMP, MCUCRSET ; set for idel
;out    MCUCR, TEMP
;sleep

; stop the flash
sbj     PORTB, LED     ; LED lamp off

; count the flashes
inc     FLASHLOW       ; Adjust Count
brne    FLASHEND
inc     FLASHLOW       ; Can not be zero
inc     FLASHHIGH      ; Adjust high byte
brne    FLASHEND
clr     FLASHLOW       ; Flash time is over stop flash
inc     FLASHHIGH      ; Set stoped

FLASHEND:
ret

ADCRUN:
; rutine for ADC
ldi     TEMP, ADMUXSET
add     TEMP, ZH
out     ADMUX, TEMP    ; Set adc chanel
sbi     ADCSR, ADSC     ; Start the ADC Conversion
;ldi    TEMP, MCUCRADDC ; set for ADC
;out    MCUCR, TEMP

;sleep
ADCRUN01:
sbis    ADCSR, ADIF     ; Test for end of conversion
rjmp    ADCRUN01        ; Loop till end
in      TEMP, ADCL      ; Get the resulats
st      Z, TEMP
inc     ZL
in      TEMP, ADCH      ; Get the resulats
st      Z, TEMP

```

FIG. 7F

LOADTEST:	ret		
	clr	LOADOK	; make load not OK
	; work load test		
	tst	TICB	; test for time to run
	bme	LOADTESTEND	; must be zero th run
	ldi	TICB, TICLOAD	; reset the timer
	sbi	DDRB, LOAD	; start Load by seting output
	; time the load		
	ldi	TEMP, TIME20m	; load timer to start
	rcall	WATE	
	;out	TCNT0, TEMP	
	;ldi	TEMP, MCUCRSET	; set for idel
	;out	MCUCR, TEMP	
	;sleep		; wate for time
	; read adc ch for SVFC		
	ldi	ZH, SVFC	; set chanel
	ldi	ZL, VFCLAN	; data store pointer
	rcall	ADCRUN	
	cbi	DDRB, LOAD	; stop Load by try staling
	; find load dif		
	mov	VFCDIFLOW, VFCBALOW	
	mov	VFCDIFHIGH, VFCBAHIGH	
	sub	VFCDIFLOW, VFCLALOW	
	sbc	VFCDIFHIGH, VFCLAGHIGH	
	; test dif		
	cpi	VFCDIFLOW, low(loaddelta)	
	ldi	TEMP, high(loaddelta)	
	cpc	VFCDIFHIGH, TEMP	
	brsh	LOADTESTEND	
LOAD10:	doc	LOADOK	; set load OK \$FF
LOADTESTEND:	ret		
	; runtime to use timer 0 for wating, Temp time		
WATE:			
	; out	TCNT0, TEMP	
	ldi	TEMP, MCUCRSET	; set for idel
	out	MCUCR, TEMP	
	sleep		; wate for time
	ret		
Trace:	; A lamp blinb rutine for testing		
	sbi	PINb, led	
	rjmp	Trace1	
	sbi	PORTb, led	
	cbi	PORTb, onnot	
	rjmp	Traceend	
Trace1:	cbi	PORTb, led	
	sbi	PORTb, onnot	
Traceend:	ret		
EXIT			

FIG. 7G

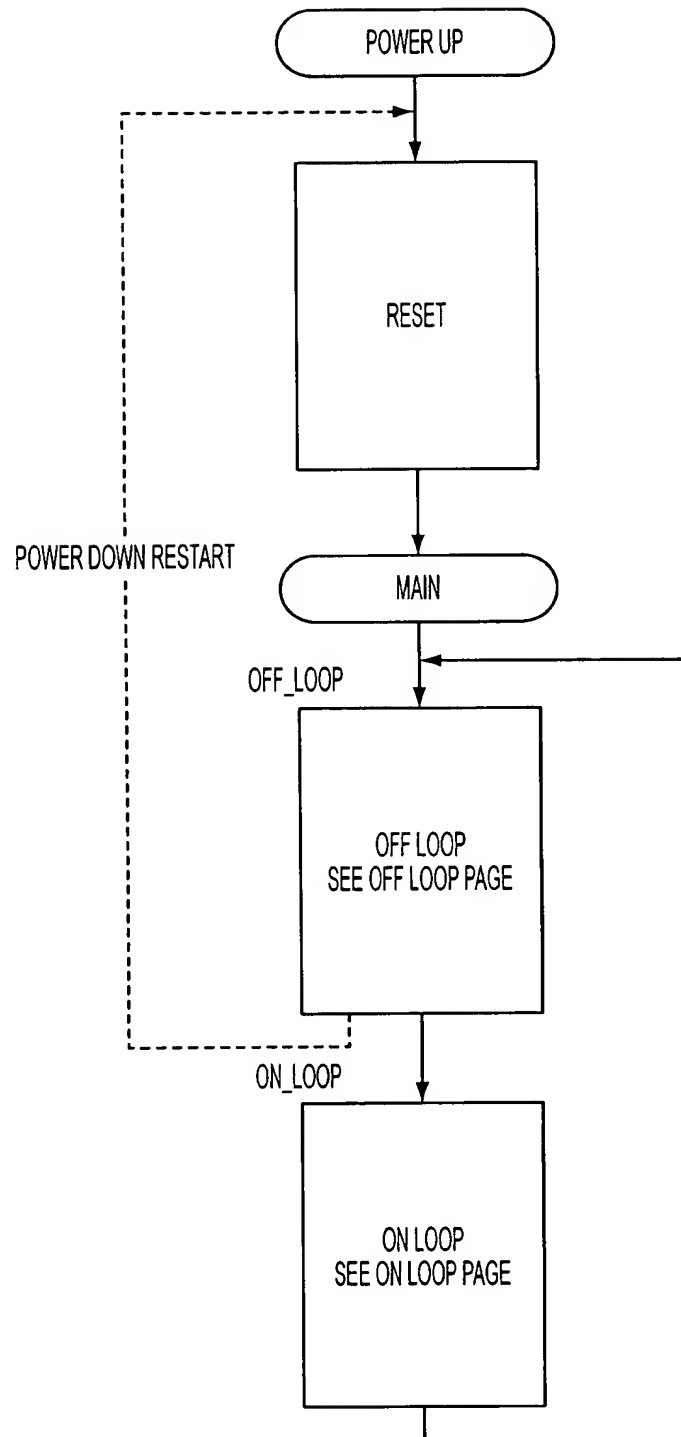


FIG. 8A

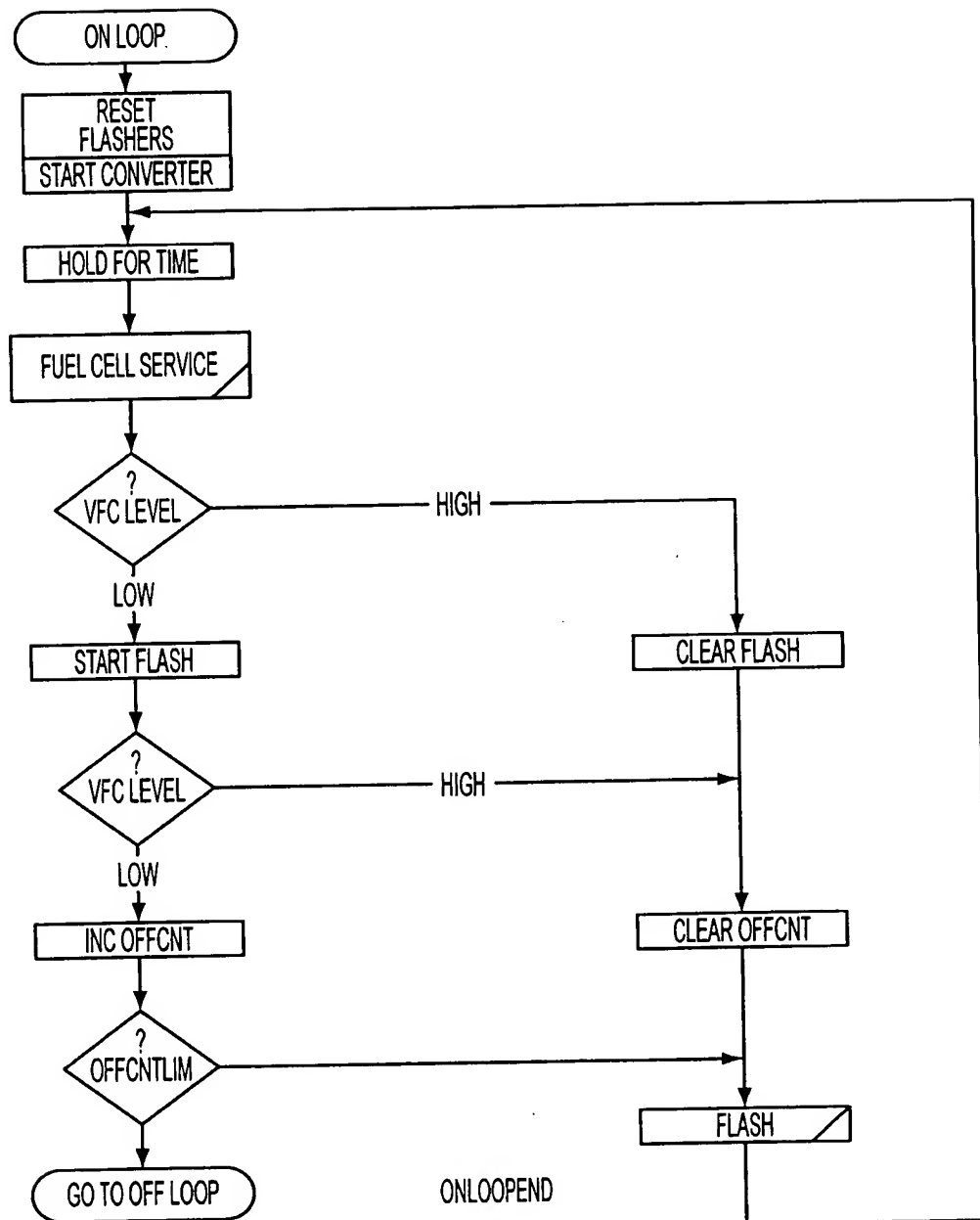


FIG. 8B

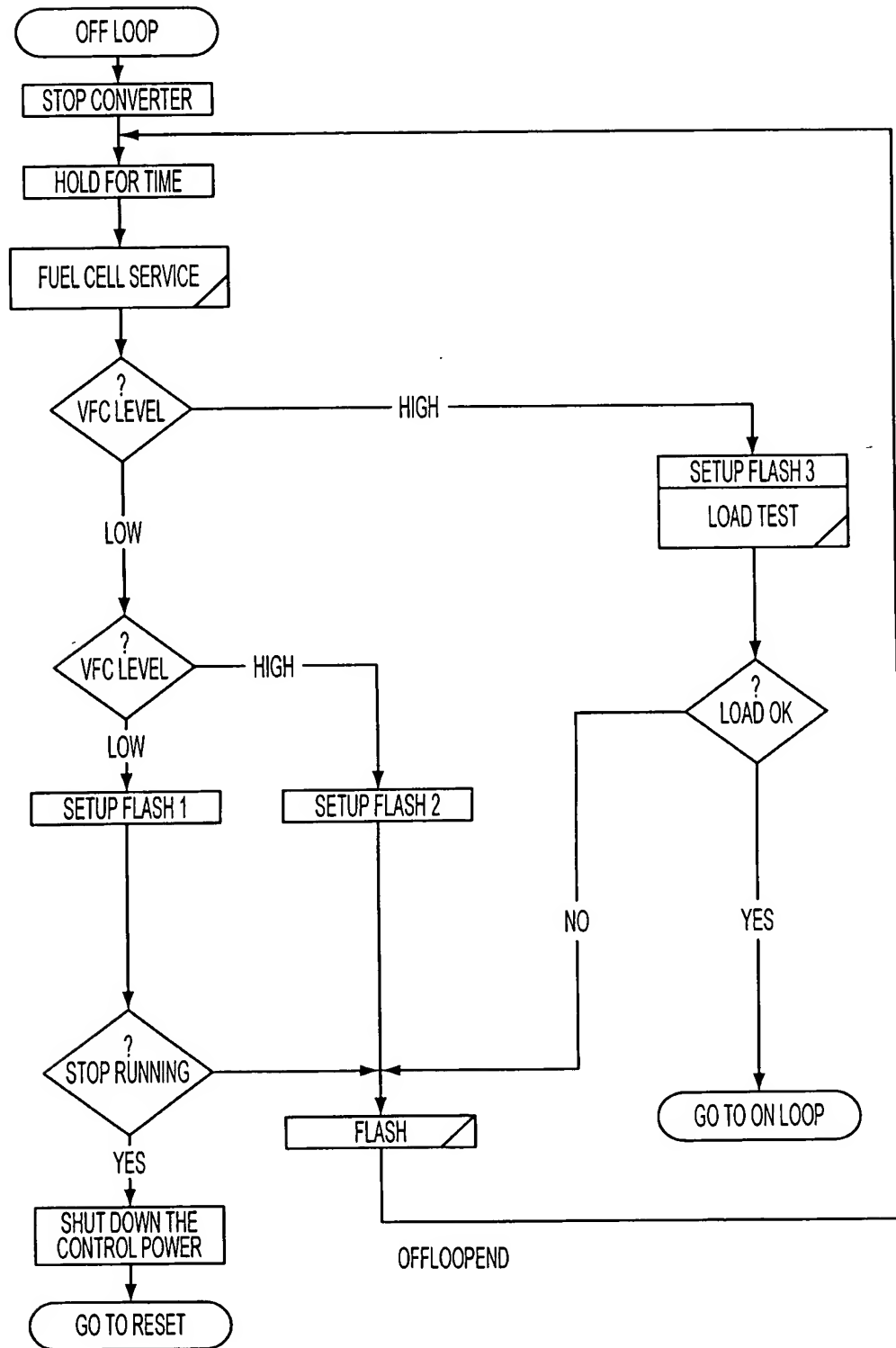


FIG. 8C

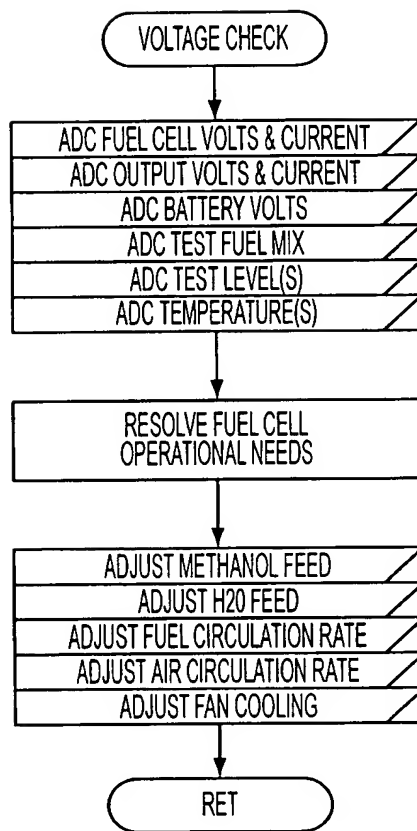


FIG. 8D

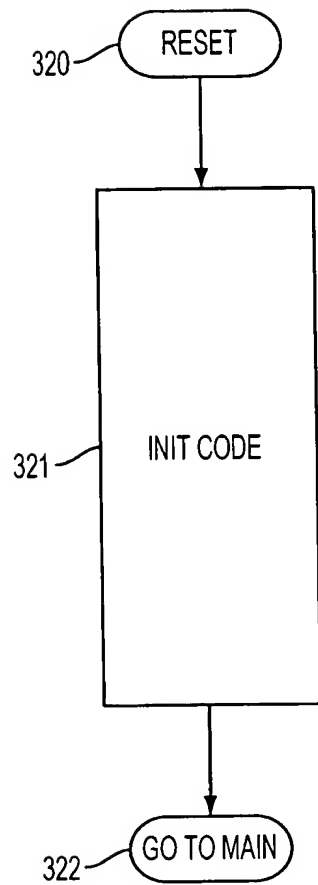


FIG. 9

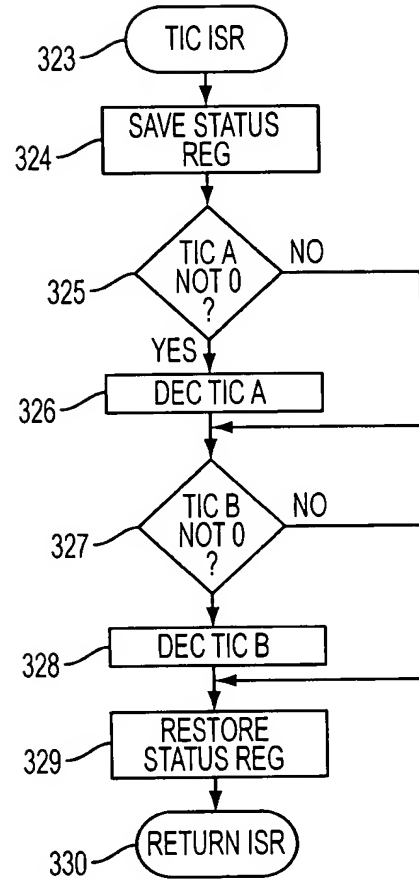


FIG. 10

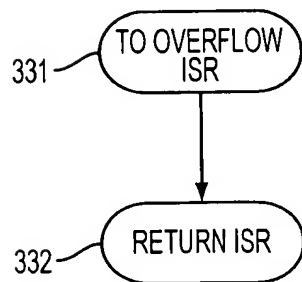


FIG. 11

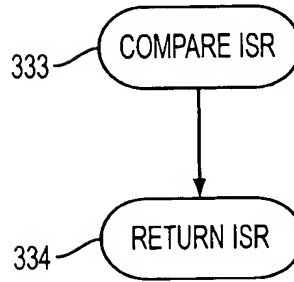


FIG. 12

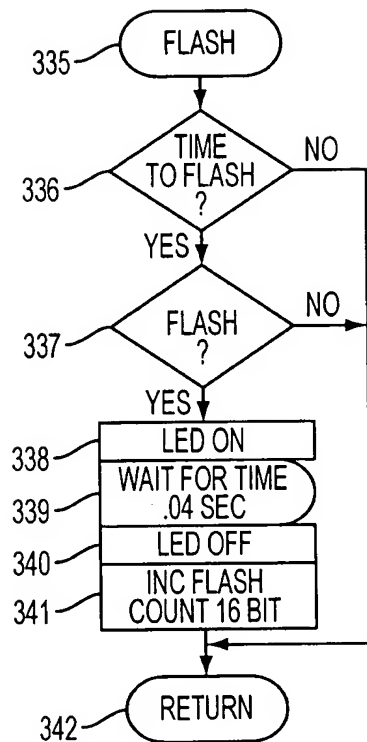


FIG. 13

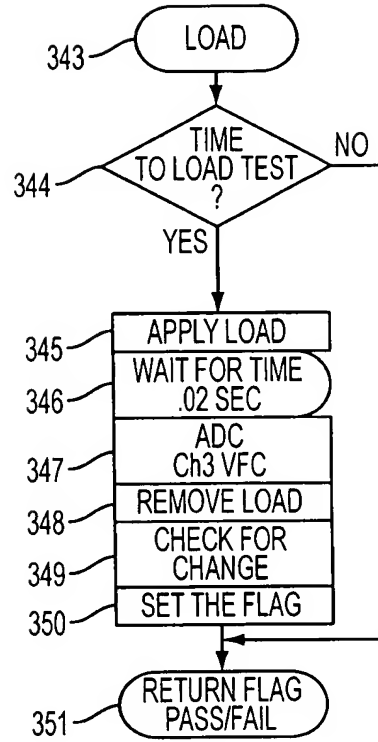


FIG. 14

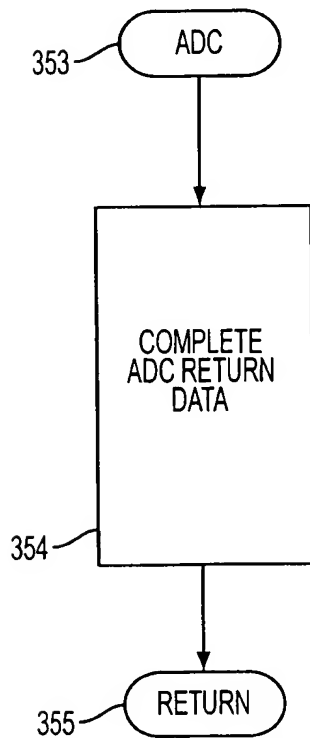


FIG. 15

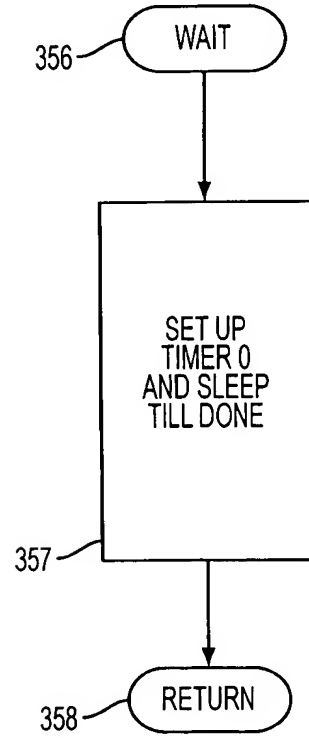


FIG. 16

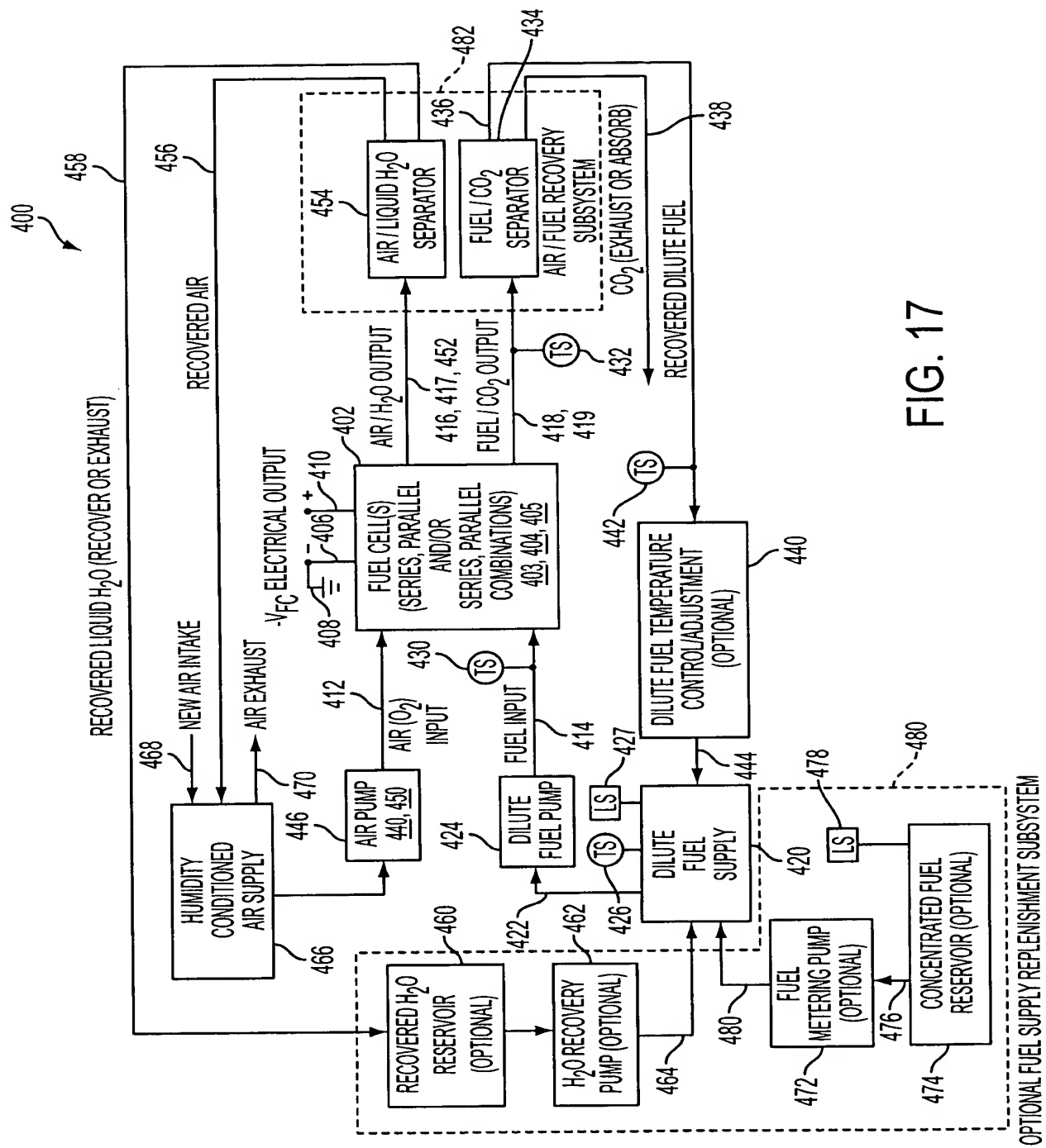


FIG. 17

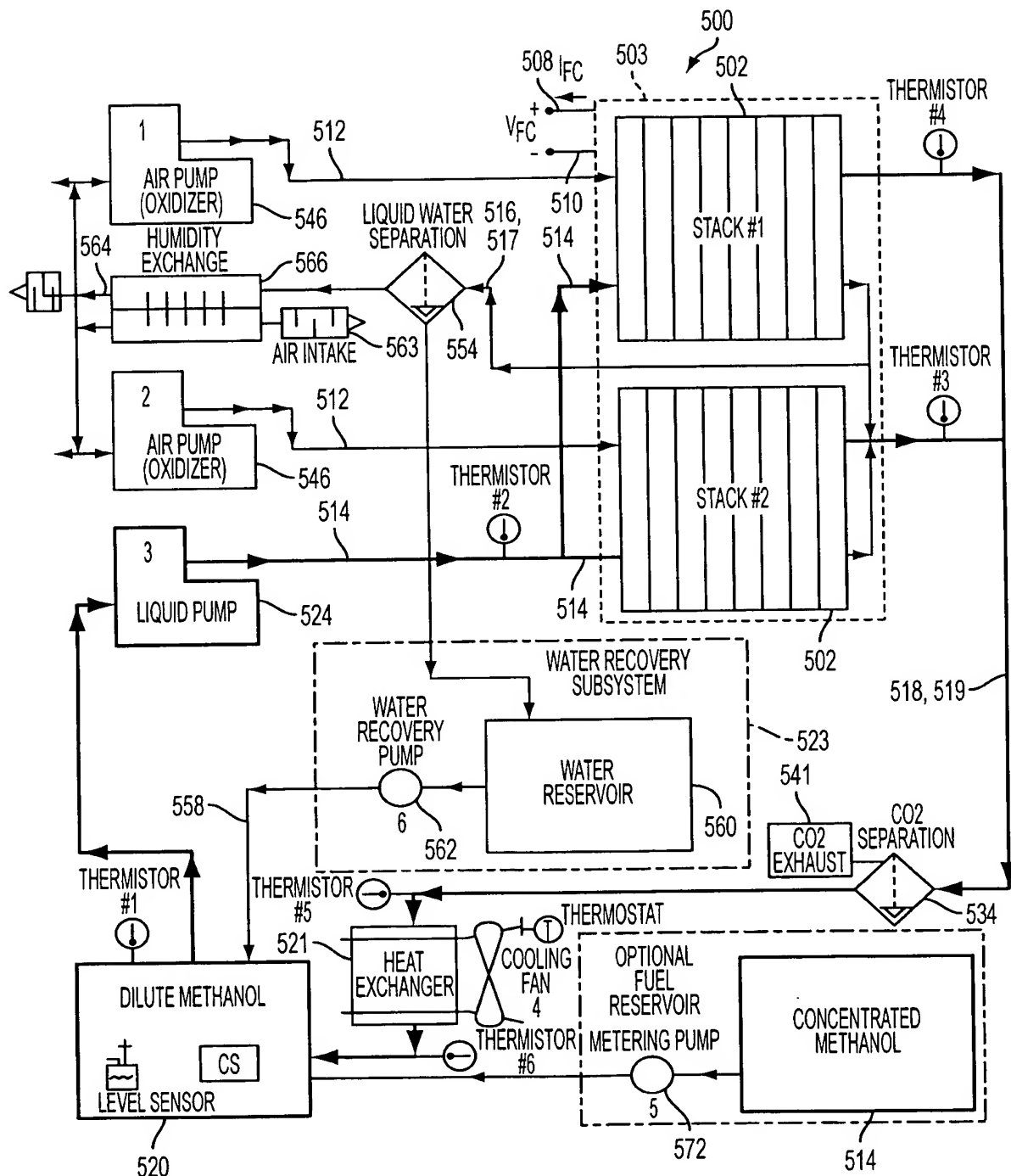


FIG. 18

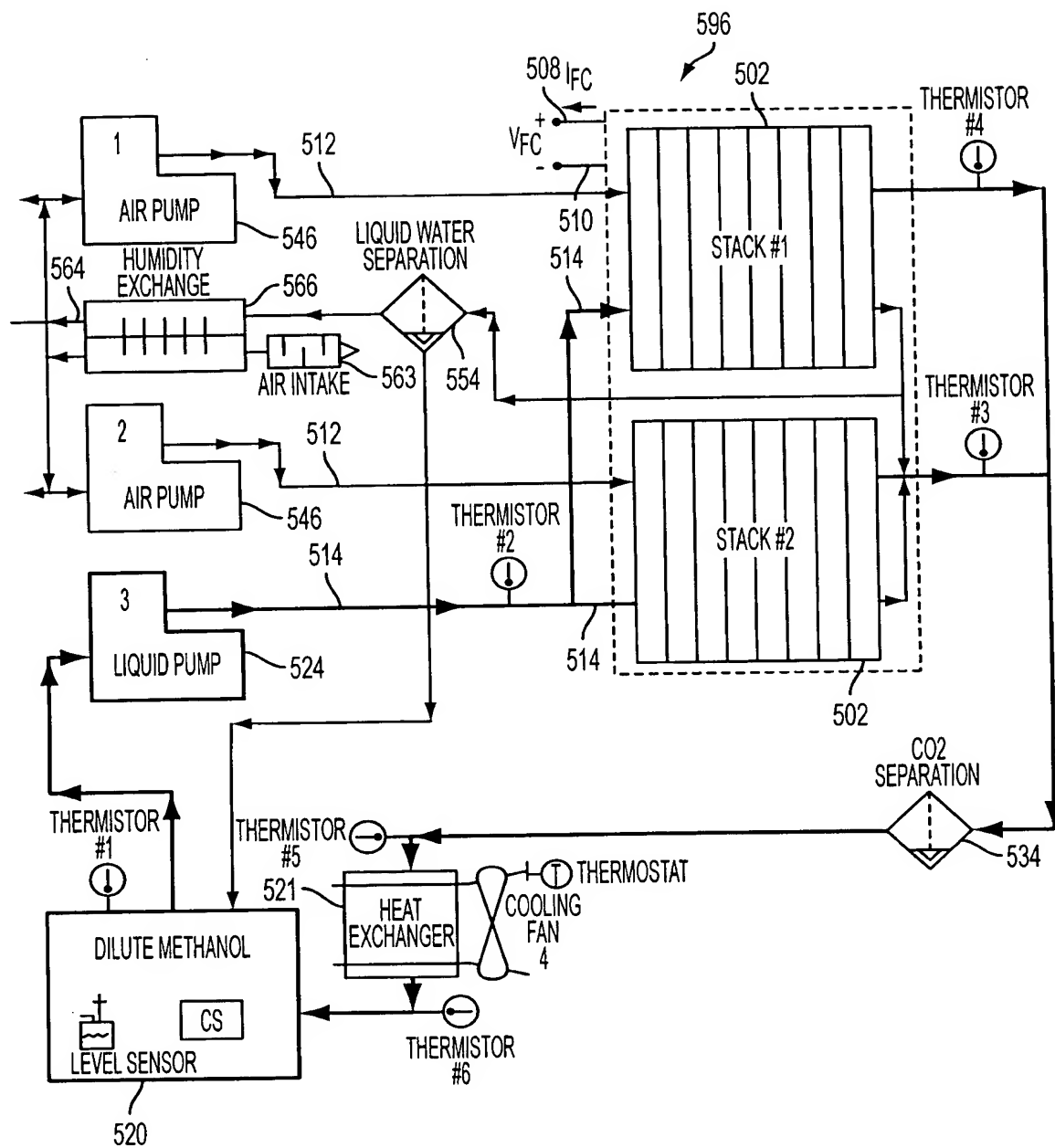


FIG. 19

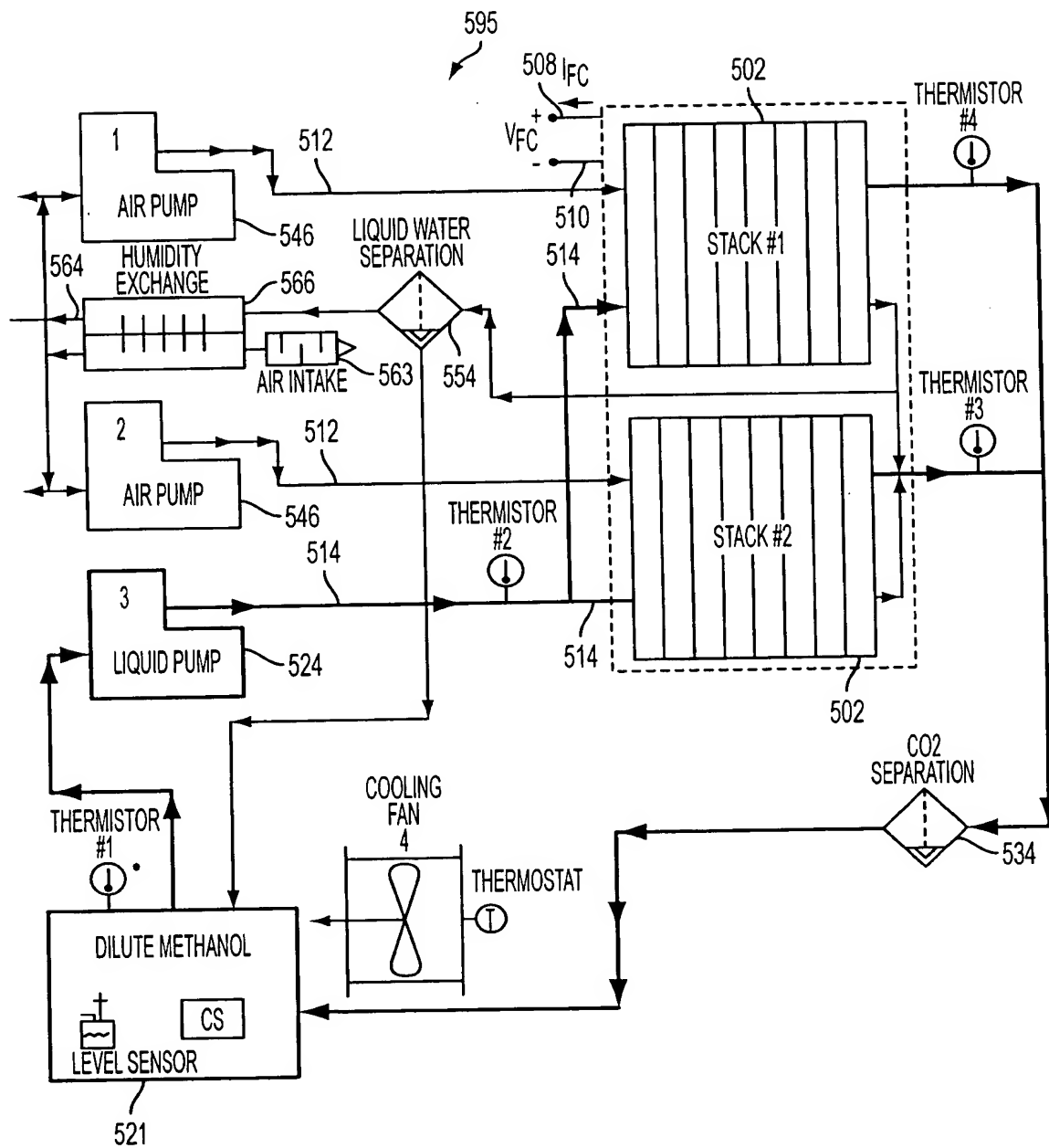


FIG. 20

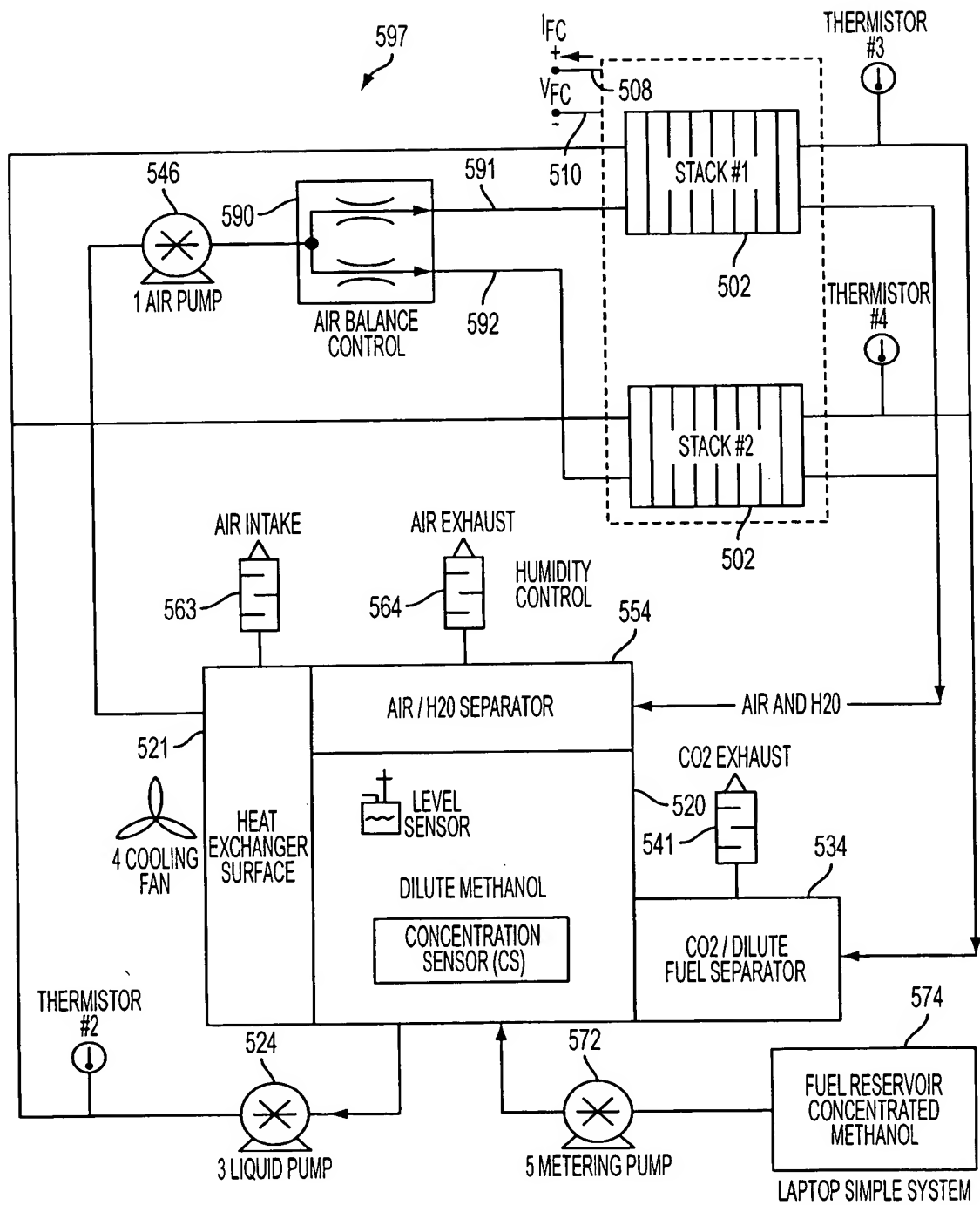


FIG. 21

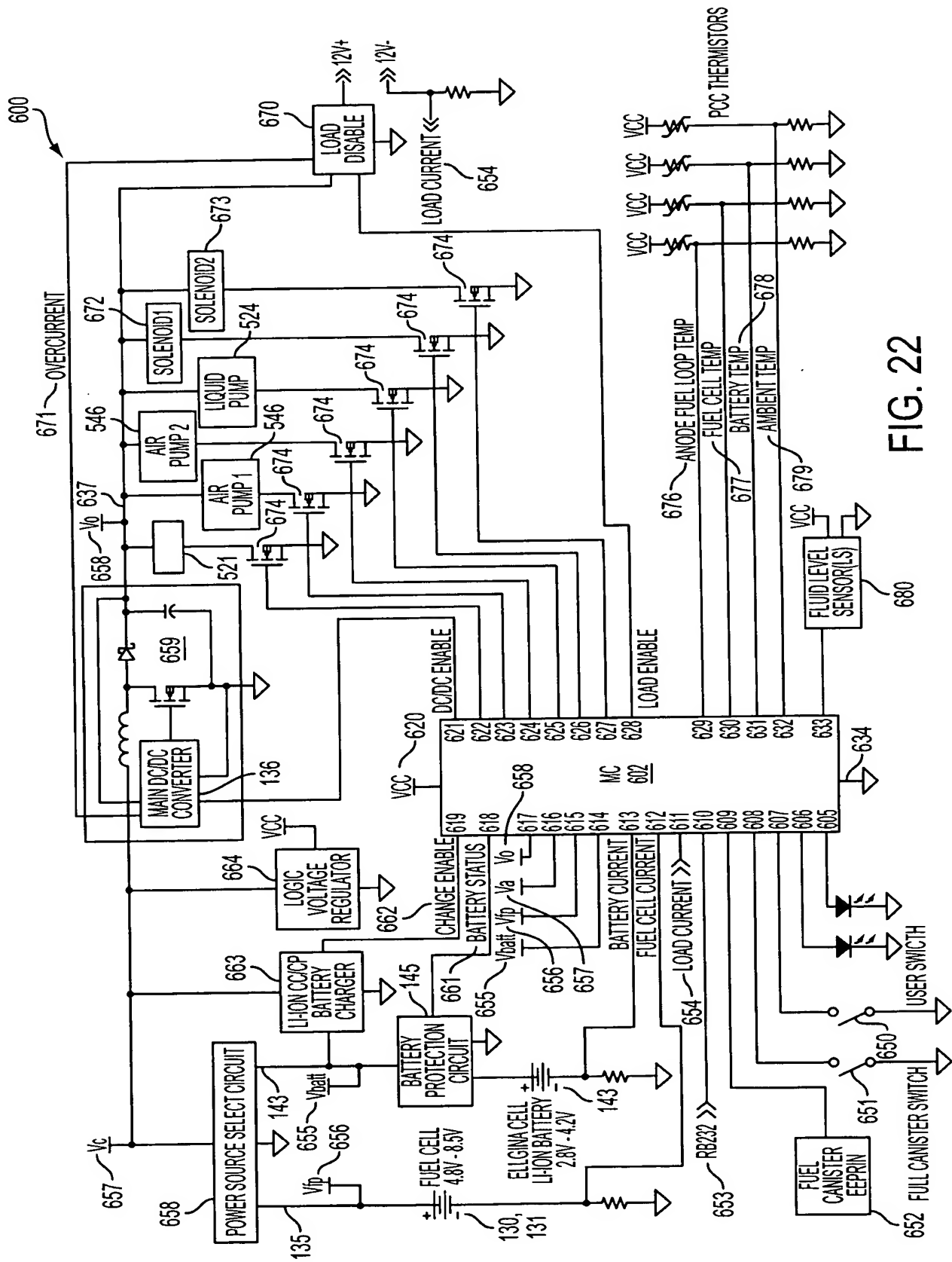


FIG. 22

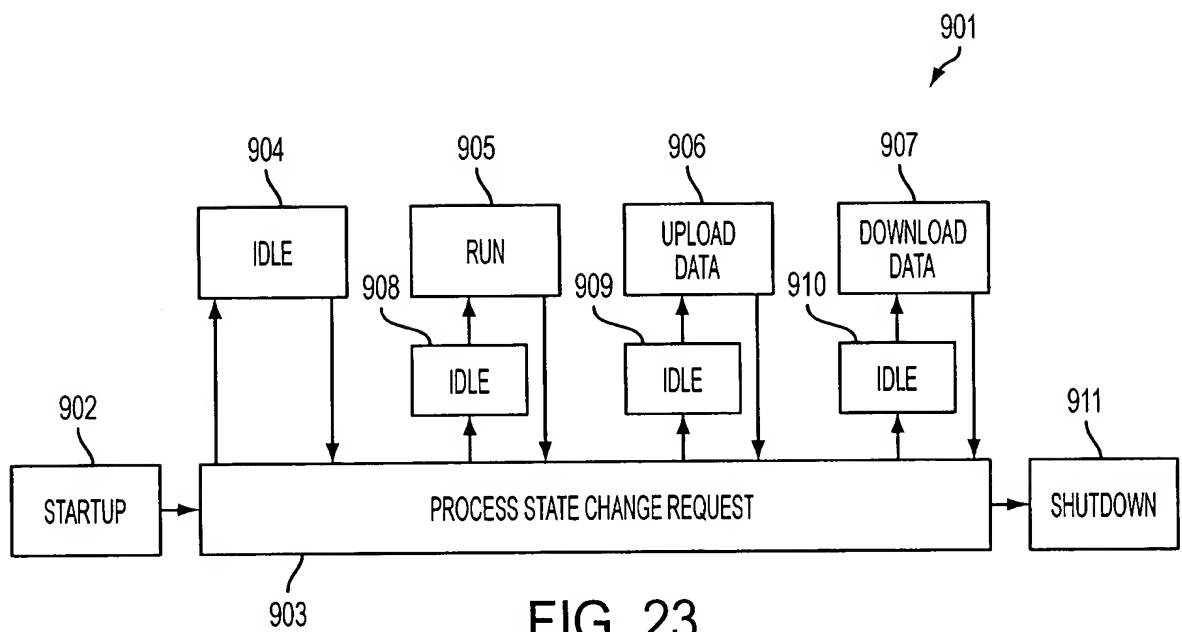


FIG. 23

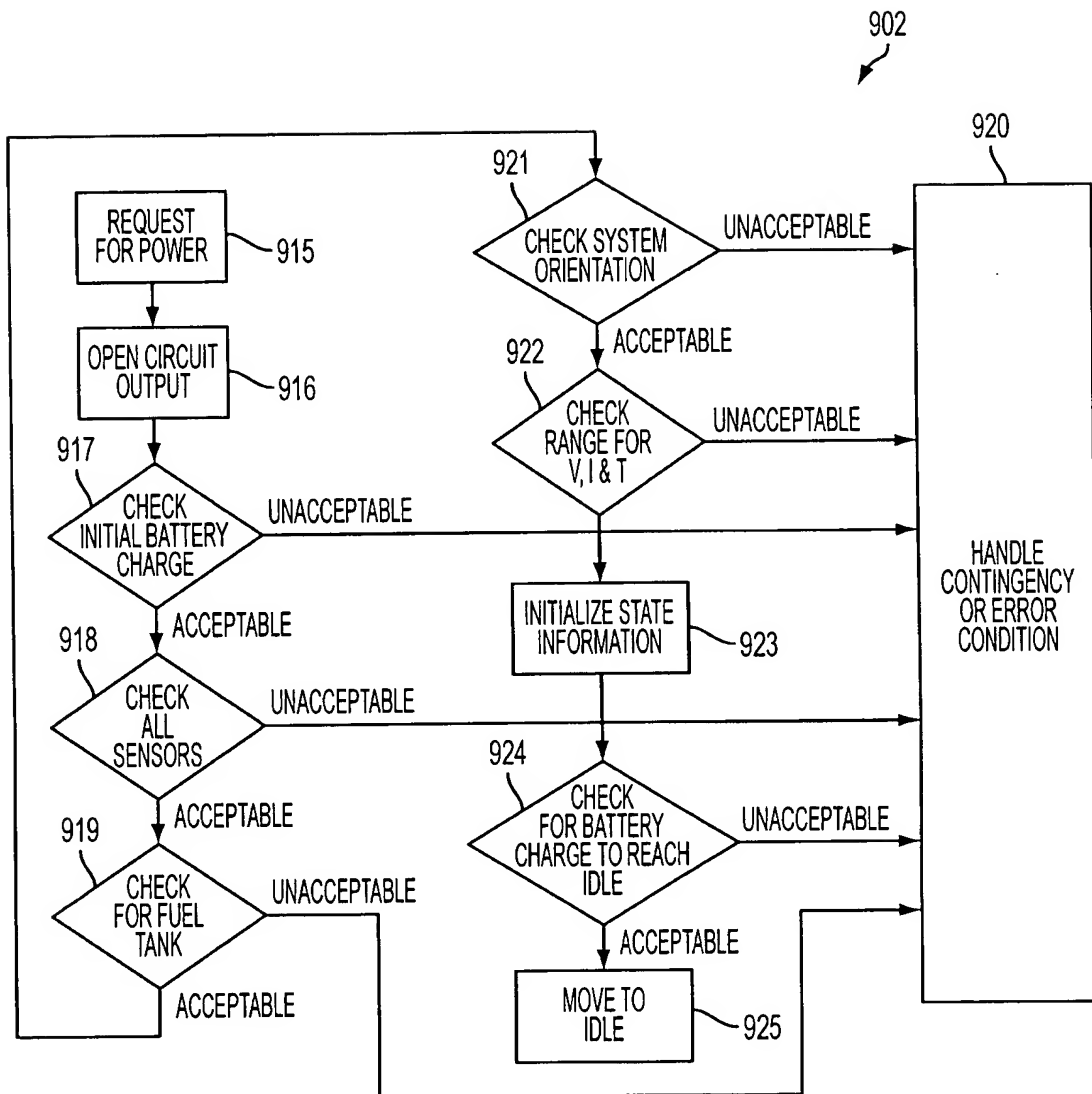


FIG. 24

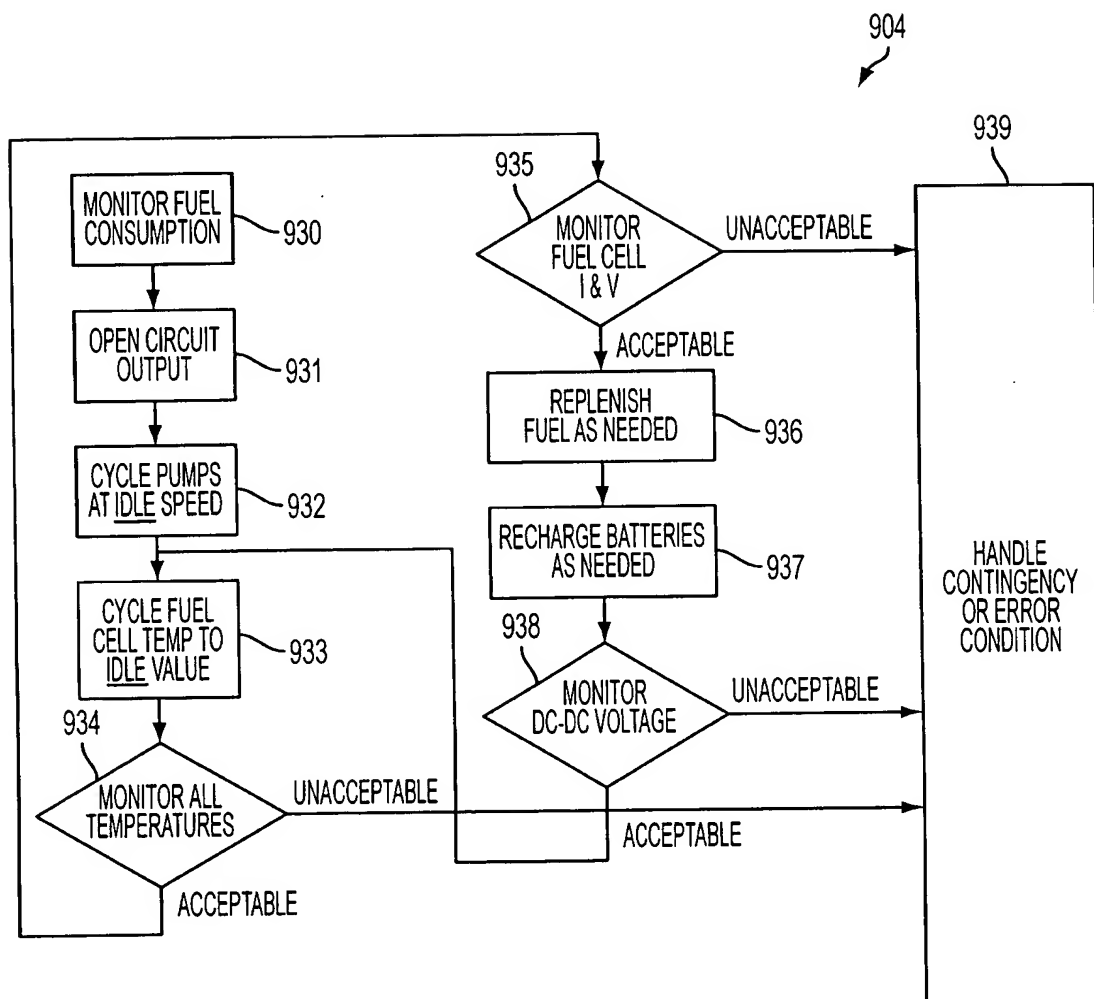


FIG. 25

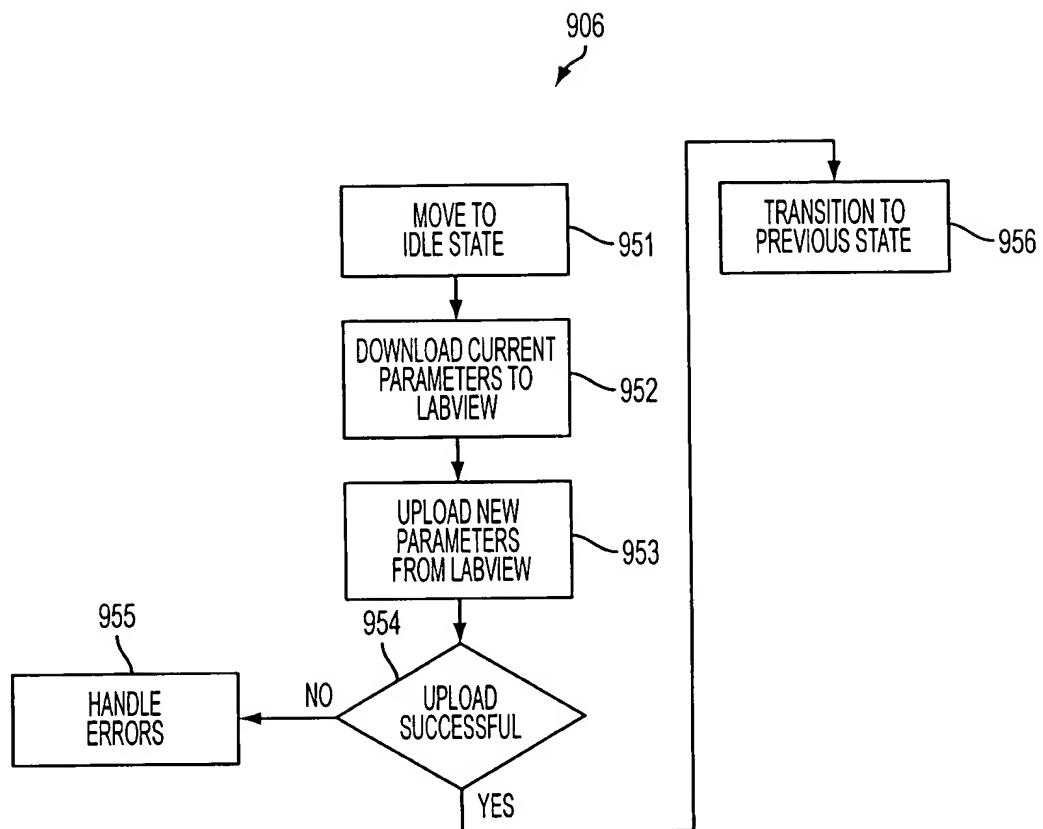


FIG. 26

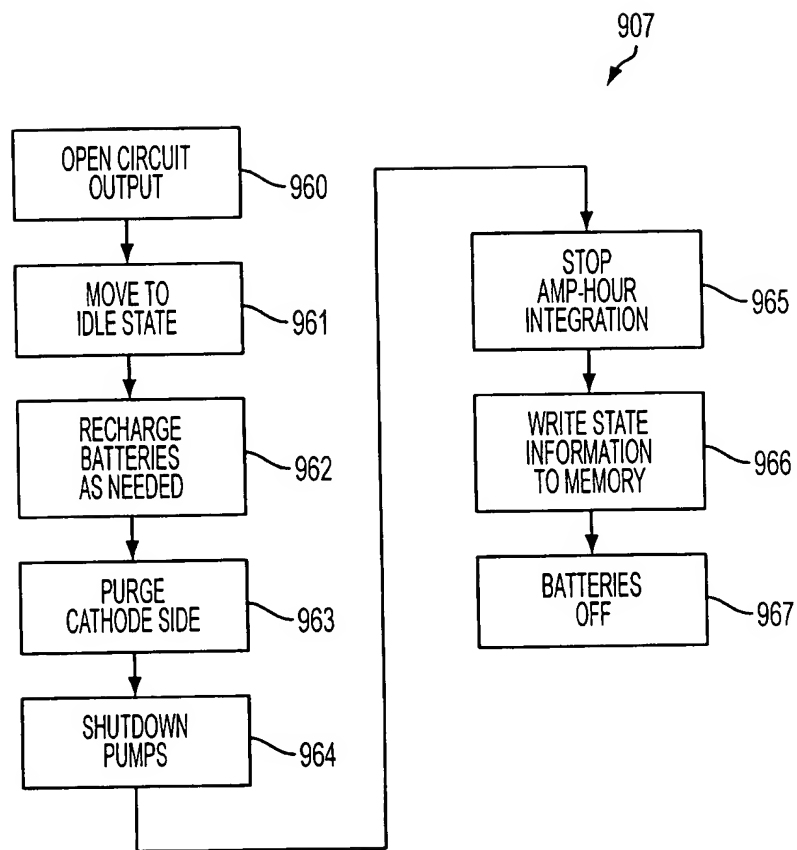


FIG. 27

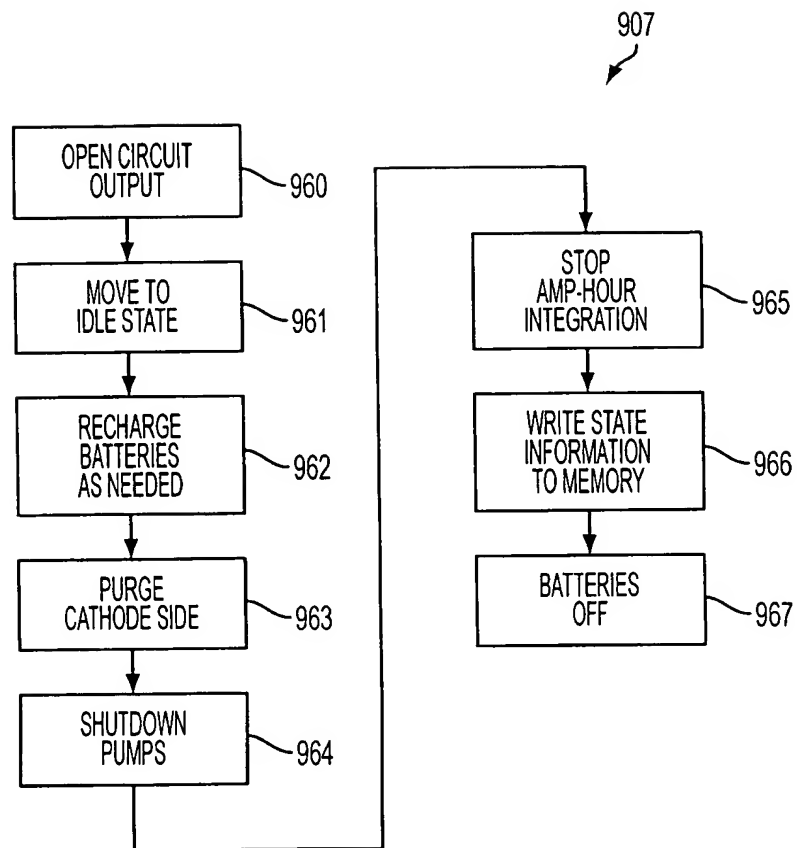


FIG. 28

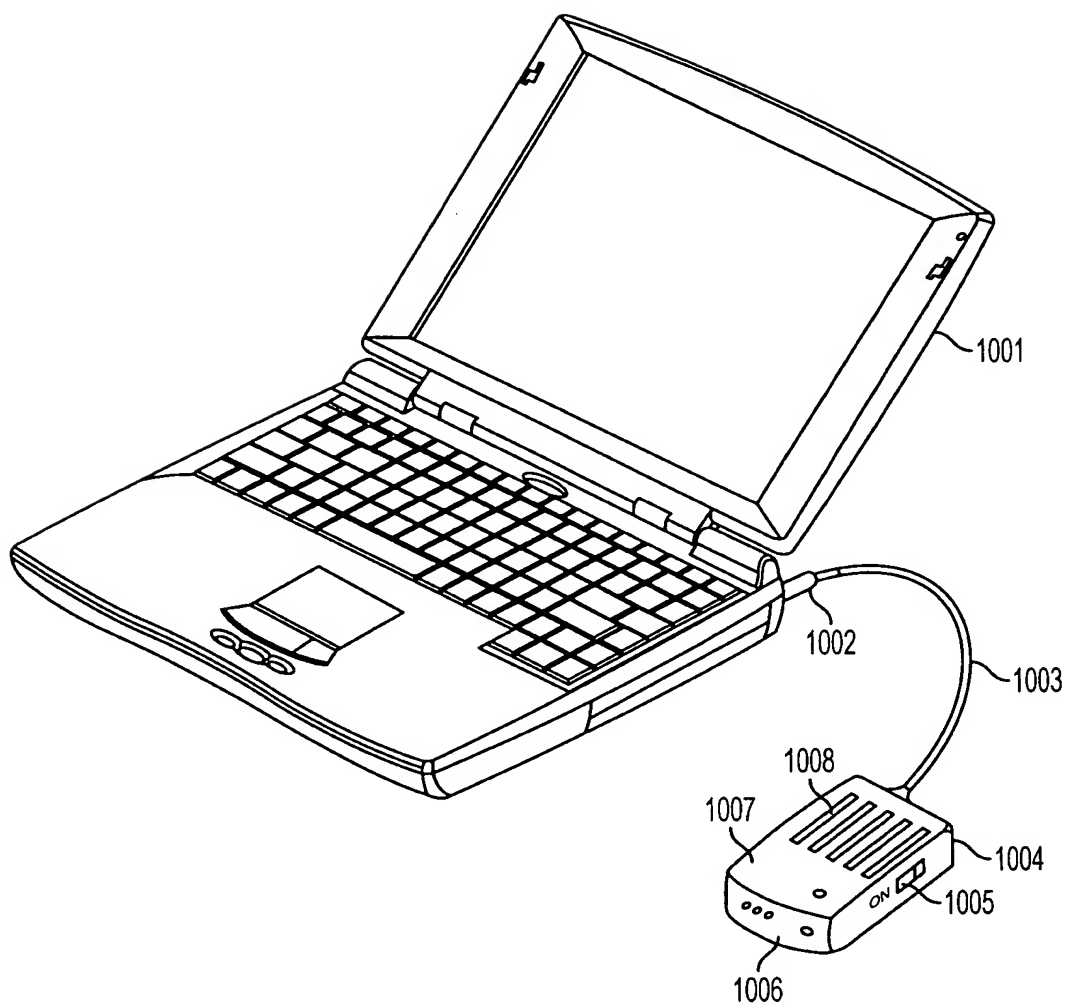


FIG. 29